

# MIB-samenstellers en MIB-laden

## Inhoud

[Inleiding](#)

[Voorwaarden](#)

[Vereisten](#)

[Gebruikte componenten](#)

[Conventies](#)

[Problemen met normale MIB-lading](#)

[Opdracht laden](#)

[Onjuiste combinaties van definities van datatype](#)

[Herdefinities van doelidentificatie](#)

[Definities van ingebouwde datatypen](#)

[Alternatieve afmetingen](#)

[Ongebruikte objecten-identificatoren](#)

[Trapdefinities](#)

[RFC 14xx-gebaseerde samenstellers versus RFC 19xx-gebaseerde samenstellers](#)

[MIBS laden en compileren in NMS van derden](#)

[Vanuit de GUI van HP OpenView of IBM NetView](#)

[Vanuit de opdrachtregel van HP OpenView of IBM NetView](#)

[Gerelateerde informatie](#)

## Inleiding

De meeste netwerkbeheersystemen bieden de gebruiker een manier om MIB's te laden. Het laden van een MIB is een manier waarop een NMS kan leren over nieuwe MIB objecten, zoals hun namen, object identifier (OIDs) en het soort datatype (bijvoorbeeld `Counter`).

Het MIB kan worden geparseerd wanneer het wordt geladen of het kan later gebeuren, bijvoorbeeld wanneer een NMS-toepassing wordt uitgevoerd. De software die het parsieren uitvoert is een MIB compiler.

Elke syntactisch correcte MIB moet met succes worden geparkeerd door de MIB-compiler van elke verkoper. Helaas kunnen verschillende MIB-samenstellers verschillende quirks vertonen.

Cisco doet voortdurende inspanningen om te verzekeren dat de MIBs die aan klanten worden gepubliceerd syntactisch correct zijn. Cisco vermijdt ook MIB-constructies die problematisch zijn gebleken in populaire NMS-producten. Ondanks deze inspanningen is het niet mogelijk om in alle MIB-samenstellers in het veld te voldoen aan de irks.

Dit document behandelt een aantal gemeenschappelijke problemen en stelt een aantal werkpunten voor. Als u een van deze problemen ondervindt met de MIB-compiler van uw verkoper (met uitzondering van de [RFC 14xx versus RFC 19xx](#) uitgifte), is dat te wijten aan een

tekortkoming in die MIB-compiler. U kunt er bij uw verkoper of verkopers op aandringen om hun samenstellers te repareren.

## Voorwaarden

### Vereisten

Lezers van dit document dienen bekend te zijn met MIB's.

### Gebruikte componenten

Dit document is niet beperkt tot specifieke software- en hardware-versies.

De informatie in dit document is gebaseerd op de apparaten in een specifieke laboratoriumomgeving. Alle apparaten die in dit document worden beschreven, hadden een opgeschoonde (standaard)configuratie. Als uw netwerk live is, moet u de potentiële impact van elke opdracht begrijpen.

### Conventies

Raadpleeg [Cisco Technical Tips Conventions](#) (Conventies voor technische tips van Cisco) voor meer informatie over documentconventies.

## Problemen met normale MIB-lading

### Opdracht laden

De ladingsvolgorde is het belangrijkste en meest voorkomende probleem wanneer u MIBs laadt. Veel MIB's gebruiken definities die in andere MIB's worden gedefinieerd. Deze definities zijn opgenomen in de `IMPORTS`-clausule in de buurt van de MIB.

Bijvoorbeeld, als MIB **mumble** een definitie van MIB **bumble** invoert, vereisen sommige MIB samenstellers dat u MIB **bumble** laadt voordat MIB **mumble** wordt geladen. Als u de ladingsvolgorde fout hebt, zal de compiler claimen dat de geïmporteerde MIB's niet gedefinieerd zijn.

Dit is een lijst van MIB's waarvan veel andere MIB's importeren, en de volgorde waarin je ze moet laden. Dit houdt waarschijnlijk rekening met 95% van de ladingsorderproblemen (de meeste andere MIB's kunnen in elke volgorde worden geladen):

- SNMPv2-SMI.my
- SNMPv2-TC.my
- SNMPv2-MIB.my
- RFC1213-MIB.my
- IF-MIB.my
- CISCO-SMI.my
- CISCO-PRODUCTS-MIB.my
- CISCO-TC.my

**Opmerking:** Als u de v1 versies van deze MIBs laadt, zal de MIB bestandsnaam er in werkelijkheid uitzien als IF-MIB-V1SMI.my ("-V1SMI" wordt toegevoegd aan de naam van MIBs die zijn geconverteerd van v2 naar v1). De uitzondering hierop is de [RFC1213-MIB.my](#) MIB, die alleen bestaat als v1 versie (dwz, er is geen RFC1213-MIB-V1SMI.my).

Als u een andere MIB probeert te laden, en als de compiler over niet gedefinieerde items klaagt, identificeer dan uit welke MIBs deze MIBs importeert en controleer of u eerst alle andere MIBs hebt geladen.

**Opmerking:** Voor elke MIB kunt u de exacte lijst zien van de MIB's die voorafgaand aan het laden moeten worden geladen—met de exacte compilatievolgorde—in [SNMP Object Navigator > View & Download MIBs](#); Selecteer **Gebiedsdelen bekijken en MIB downloaden**.

## [Onjuiste combinaties van definities van datatype](#)

Hoewel de definities van Cisco MIB datatype niet verkeerd zullen worden afgestemd, kunt u dat in het geval voor sommige standaard RFC MIBs vinden. Bijvoorbeeld:

- MIB-**mumble** definieert: `AnyDatatype := INTEGER(0.100)`
- MIB **bumble** definieert: `AnyDatatype := INTEGER(1.50)`

Dit voorbeeld wordt beschouwd als een triviale fout en de MIB ladingen succesvol met een waarschuwingsbericht.

Het volgende voorbeeld wordt beschouwd als een niet-triviale fout (hoewel de twee definities in wezen gelijkwaardig zijn), en de MIB wordt niet succesvol geparkeerd.

- MIB-**mumble** definieert: `SommigeDatatype := DisplayString`
- MIB **bumble** definieert: `someDatatype := OCTET STRING (SIZE (0.255))`

Als uw MIB compiler deze als fouten behandelt, of als u de waarschuwingsberichten wilt weghalen, dan bewerk dan één van de MIBs die dit zelfde gegevenstype bepalen zodat de definities overeenkomen.

## [Herdefinities van doelidentificatie](#)

U kunt OID opnieuw definiëren als u deze MIB's laadt (hoewel er andere gevallen kunnen zijn waarin deze fout optreedt):

- [OUD-CISCO-CPU-MIB.my](#)
- [OUD-CISCO-ENV-MIB.my](#)
- [OUD-CISCO-MEMORY-MIB.my](#)
- [OUD-CISCO-SYSTEM-MIB.my](#)

Bijvoorbeeld:

- OLD-CISCO-CPU-MIB.my definieert: `lcpu OBJECT IDENTIFIER := { Local 1 }`
- OUD-CISCO-ENV-MIB.my definieert: `lenv OBJECT IDENTIFIER := {1}`

Wanneer u deze twee MIBs laadt, kan de MIB compiler klagen over het `lcpu OBJECT IDENTIFIER` die met een nieuwe naam `lenv` wordt herdefinieerd. De OLD-CISCO-MEMORY-MIB.my en OLD-CISCO-SYSTEM-MIB.my geven ook nieuwe namen aan `< plaatselijk 1`.

Dit wordt behandeld als een triviale fout en de MIB laadt met succes met een

waarschuwingsbericht.

Als de MIB niet succesvol laadt, of als u het waarschuwingsbericht wilt verwijderen, bewerk dan een van de MIB's zodat alle MIB's dezelfde naam gebruiken.

## Definities van ingebouwde datatypen

Vele MIB samenstellers hebben ingebouwde kennis van sommige gegevenstypen, zoals DisplayString. Sommige van deze samenstellers klagen of ze een definitie zien voor deze data in een MIB. DisplayString is bijvoorbeeld gedefinieerd in SNMPv2-TC.

De tijdelijke versie is het verwijderen of becommentariëren van de definitie in het MIB-bestand.

## Alternatieve afmetingen

Dit is een syntactisch geldig voorbeeld, dat aangeeft dat een waarde van type `MyDatatype` lang 0, 5 of 20 octetten zal zijn:

```
MyDatatype ::= OCTET STRING (SIZE(0 | 5 | 20))
```

Sommige MIB-samenstellers accepteren deze syntaxis niet. Normaal gesproken is het voldoende om een van de groottes te kiezen en de andere te verwijderen. Je moet de grootste grootte houden. Zo zou het vorige voorbeeld worden gewijzigd:

```
MyDatatype ::= OCTET STRING (SIZE(20))
```

## Ongebruikte objecten-identificatoren

Sommige OID's worden als vreemd beschouwd omdat ze niet verwijzen naar een knooppunt in het SMI (zoals de meeste OBJECT-IDENTIFIER's). Ze zijn echter syntactisch geldig. Een algemeen voorbeeld is de ongeldige object identifier, bijvoorbeeld, `< 0 >`. Sommige MIB-samenstellers geven niet om OBJECT-IDENTIFIER's die niet corresponderen met een knooppunt in de SMI. Dit zijn voorbeelden van de syntaxis van MIB die problemen voor deze samenstellers kan veroorzaken:

```
zeroDotZero OBJECT IDENTIFIER ::= { 0 0 }  
myMIBObject OBJECT-TYPE  
DEFVAL { {0 0} }
```

De tijdelijke versie is het verwijderen of becommentariëren van deze soorten verwijzingen in het MIB-bestand.

## Trapdefinities

In SNMPv1 MIBs worden de vallen gedefinieerd met de macro `TRAP-TYPE`. In SNMPv2 MIBs worden de vallen gedefinieerd met behulp van de macro van het `aanmeldingstype`.

Sommige MIB samenstellers houden niet van deze definities in de MIB dossiers die zij ontleden (zij steunen deze macro's niet).

Als dit het geval is, kunt u de valdefinities verwijderen of de definities uitcommenteren (bijvoorbeeld, leg de MIB commentaar delimiter – aan het begin van de regels).

## [RFC 14xx-gebaseerde samenstellers versus RFC 19xx-gebaseerde samenstellers](#)

RFC's 1442 tot en met 1452 definiëren de op partijen gebaseerde SNMPv2. Deze RFC's zijn verouderd door de nieuwere ontwerp-standaard RFC's van 1902 tot en met 1908.

Wat de syntaxis van het MIB betreft, zijn er zeer weinig verschillen tussen deze twee versies van SNMPv2; er zijn echter enkele . Cisco MIBs zijn momenteel gebaseerd op RFC 19xx-regels.

**Opmerking:** Een paar jaar geleden, toen Cisco MIBs RFC 14xx-gebaseerd waren, klaagden sommige op RFC 19xx gebaseerde samenstellers over de `Unsigned32 := TEXTUAL-CONVENTION`-lijn in CISCO-TC.my en PNNI-MIB.my MIBs. Dit komt doordat Unsigned32 een vooraf gedefinieerd datatype is in RFC 19xx. Om deze reden, gebruikte Cisco om alternatieve versies van deze MIBs (CISCO-TC-NO-U32.my en PNNI-MIB-NO-U32.my) te hebben zonder definitie voor Unsigned32, om in de samenstellers te laden die al van dit gegevenstype weten. Dit is niet langer van toepassing.

## [MIBS laden en compileren in NMS van derden](#)

De beste en meest efficiënte manier om Cisco MIBs, vallen en pictogrammen in NMS van derden te laden is het gebruik van CiscoWorks Integration Utility (Integration Utility), dat beschikbaar is als deel van CiscoWorks Common Services (of standalone van <http://www.cisco.com/cgi-bin/tablebuild.pl/cw2000-utility>), met de corresponderende adapter voor integratiehulpprogramma's van <http://www.cisco.com/tacpage/sw-center/cw2000/cmc3rd.shtml> en het nieuwste Data Bundle voor netwerkbeheer (NMIDB). Controleer de documentatie bij het integratiehulpprogramma voor meer informatie.

U kunt ook de documentatie van NMS van derden over het laden en compileren van MIB raadplegen. Dit document bevat instructies voor HP OpenView en IBM NetView. Maar u dient nog steeds HP of IBM documentatie te raadplegen, omdat de producten kunnen veranderen.

## [Vanuit de GUI van HP OpenView of IBM NetView](#)

Volg deze stappen om de Cisco MIBs te laden die u wilt:

1. Kopieer de bestanden naar de directory/usr/OV/snmp\_mibs van het netwerkbeheerstation. Dit is de standaardmap waarin HP OpenView en IBM NetView naar MIB documenten zoeken. Als u deze ergens anders plaatst, specificeert u de expliciete padnamen in de grafische interface van het bestand.
2. Stel de permissies in zodat u toegang tot de MIB's hebt gelezen.
3. Selecteer in het menu GUI de optie **Opties > MIB's laden/verwijderen**.
4. Volg de instructies in de platformdocumentatie, om de Cisco MIBs te compileren of laden.

## [Vanuit de opdrachtregel van HP OpenView of IBM NetView](#)

Geef de opdracht `/opt/OV/bin/xmlloadmib - load filename` uit om het MIB-bestand te laden.

## [Gerelateerde informatie](#)

- [Technische ondersteuning en documentatie – Cisco Systems](#)